

IDS #  
(33-700737) (us)

# 中華民國專利公報 [19] [12]

[11]公告編號：533377

[44]中華民國 92年 (2003) 05月 21日

發明

全 11 頁

[51] Int.Cl<sup>07</sup> : G06F9/45

[54]名稱：在網路環境中用以參數擴展、產生及執行原稿之方法、系統及程式

[21]申請案號：090107602

[22]申請日期：中華民國 90年 (2001) 03月 30日

[30]優先權：[31]09/575,773

[32]2000/05/22

[33]美國

[72]發明人：

珊迪普 A. 阿明 美國  
布萊恩 D. 尼爾森 美國  
明柳嫻 美國

[71]申請人：

萬國商業機器公司 美國

[74]代理人：陳長文 先生

1

2

[57]申請專利範圍：

1.一種用以在網路環境中產生原稿之方法，包含：

以一程式設計語言來撰寫一外殼原稿，其中該外殼原稿是用於一網路環境；

以程式來轉換該外殼原稿成為一原稿，且該原稿可在一特定平台之下執行；與  
在該特定平台之下執行該原稿。

2.如申請專利範圍第1項之方法，其中以程式來轉換外殼原稿進一步包含：

定義一原稿產生指令於外殼原稿；

定義該原稿產生指令之參數；與

利用該等參數來執行該原稿產生指

令以產生該原稿。

3.如申請專利範圍第2項之方法，其中定義該原稿產生指令之參數進一步包含：

5. 定義具有相關之前置變數之前置名稱，其中該等前置名稱傳送至該原稿產生指令；與  
取得該原稿產生指令之前置名稱與前置變數。

10. 4.如申請專利範圍第3項之方法，其中定義該原稿產生指令之參數進一步包含：

自一表擷取該等前置變數之值。

5.如申請專利範圍第4項之方法，其中

15. 自一表擷取該等前置變數之值進一

步包含：

自該表擷取用以定義該等前置變數之引數與旗標。

- 6.如申請專利範圍第2項之方法，其中定義一原稿產生指令進一步包含：
  - 定義具有至少一節名稱，至少一物件型式，與至少一前置號碼之該原稿產生指令。
- 7.如申請專利範圍第2項之方法，其中以一程式設計語言來撰寫一用於網路環境之外殼原稿進一步包含：
  - 以該程式設計語言來撰寫具有該外殼原稿之一檔案；且
  - 藉由使用該程式設計該語言之一編譯程式來編譯該檔案，以產生該原稿產生指令。
- 8.一種用以在網路環境中產生原稿之系統，包含：
  - 一具有至少一伺服器系統與一從屬系統之網路環境，其中該至少一伺服器系統與該從屬系統彼此耦接以進行通訊；
  - 其中以一程式設計語言來撰寫一外殼原稿，且該外殼原稿是用於一網路環境；
  - 其中該從屬系統以程式來轉換該外殼原稿成為一原稿，且該原稿可在該伺服器系統之一特定平台之下執行；與
  - 其中伺服器系統在該特定平台之下執行該原稿。
- 9.如申請專利範圍第8項之系統：
  - 其中一原稿產生指令定義於外殼原稿；
  - 其中該原稿產生指令之參數受到定義；且
  - 其中伺服器系統接收及利用該等參數來執行該原稿產生指令，以產生該原稿。
- 10.如申請專利範圍第9項之系統，其

中：

該原稿產生指令之前置名稱受到定義，其中該等前置名稱具有相關之前置變數；且

5. 伺服器系統取得該原稿產生指令之前置名稱與前置變數。
- 11.如申請專利範圍第10項之系統，其中：
  - 伺服器系統自一表擷取該等前置變數之值。
10. 12.如申請專利範圍第11項之系統，其中：
  - 伺服器系統自該表接收用以定義該等前置變數之引數與旗標。
15. 13.如申請專利範圍第9項之系統，其中：
  - 具有至少一節名稱，至少一物件型式，與至少一前置號碼之該原稿產生指令受到定義。
20. 14.如申請專利範圍第9項之系統，其中：
  - 以該程式設計語言來撰寫具有該外殼原稿之一檔案；且
  - 藉由使用該程式設計語言之一編譯程式來編譯該檔案，以產生該原稿產生指令。
25. 15.一種程式產品，該程式產品具有一用以在網路環境中產生原稿之控制程式，該控制程式包含：
  - 該控制程式具有以一程式設計語言來撰寫之一外殼原稿，其中該外殼原稿是用於一網路環境；
  - 該控制程式以程式來轉換該外殼原稿成為一原稿，且該原稿可在一特定平台之下執行；
  - 該控制程式在該特定平台之下執行該原稿；與
  - 承載該控制程式之電腦可用型媒體。
30. 16.如申請專利範圍第15項之程式產
35. 40.

品，其中該控制程式以程式來轉換外殼原稿進一步包含：

該控制程式定義一原稿產生指令於外殼原稿；

該控制程式定義該原稿產生指令之參數；與

該控制程式利用該等參數來執行該原稿產生指令，以產生該原稿。

17. 如申請專利範圍第 15 項之程式產品，其中該控制程式定義該原稿產生指令之參數進一步包含：

該控制程式定義具有相關之前置變數之前置名稱，其中該等前置名稱傳送至該原稿產生指令；與

該控制程式取得該原稿產生指令之前置名稱與前置變數。

18. 如申請專利範圍第 17 項之程式產品，其中該控制程式定義該原稿產生指令之參數進一步包含：

該控制程式自一表擷取該等前置變數之值。

19. 如申請專利範圍第 18 項之程式產品，其中該控制程式自一表擷取該等前置變數之值進一步包含：

該控制程式自該表擷取用以定義該等前置變數之引數與旗標。

20. 如申請專利範圍第 16 項之程式產品，其中該控制程式定義一原稿產生指令進一步包含：

該控制程式定義具有至少一節名稱，至少一物件型式，與至少一前置號碼之該原稿產生指令。

21. 如申請專利範圍第 16 項之程式產品，其中該控制程式具有以一程式設計語言來撰寫之一用於網路環境之外殼原稿進一步包含：

該控制程式具有一檔案，且該檔案具有以該程式設計語言來撰寫之該外殼原稿；與

該控制程式指示以該程式設計語言

之一編譯程式來編譯該檔案，以產生該原稿產生指令。

圖式簡單說明：

圖 1 是一根據以前技藝之網路環境之方塊圖，以展示如何輸入參數至從屬系統，與如何傳送針對一選定物件之一選定指令之原稿自從屬系統至伺服器系統；

圖 2A 是一根據本發明之網路環境之方塊圖，以展示如何使用物件於一原稿之參數擴展，產生，與執行；

圖 2B 是一示範資料處理系統之硬體方塊圖，其中該示範資料處理系統可為伺服器系統或從屬系統，且是由本發明來使用；

圖 3 展示根據本發明之原稿產生指令(例如 "Ecommand")之一通用格式或語法；

圖 4 是一展示本發明之執行範例之流程圖，其中藉由使用物件來在一網路環境中提供一原稿之參數擴展，產生與執行；

圖 5A 是一用於一原始原稿產生指令檔案之程式碼樣本，其中該程式碼樣本是用以根據本發明來開發該原稿產生指令(例如 "Ecommand")；

圖 5B 是一示範與簡化之 JAVA 使用者物件類別，其中該 JAVA 使用者物件類別表示使用者物件，撰寫於伺服器系統，且是根據本發明在一網路環境中使用物件來達成一原稿之參數擴展，產生與執行；

圖 5C 是一利用圖 5A 之原始指令檔案來產生之示範 JAVA 指令檔案，其中產生之 JAVA 檔案是藉由一編譯程式來編譯，以根據本發明來產生原稿產生指令(例如 "Ecommand")；

圖 5D 是一原稿範例，其中該原稿範例是，根據本發明，在原稿產生指令(例如 "Ecommand")完成參數評估與

(4)

7

8

擴展之後受到產生：

圖6是一示範表(例如雜湊表)，其中該表是根據本發明用以產生一原稿與對應之引數，旗標與值；且

圖7是一用以執行本發明之原稿 5.

產生指令(例如 "Ecommand")之演算法範例的流程圖，其中該演算法在一網路環境中使用物件來提供原稿之參數擴展，產生與執行。

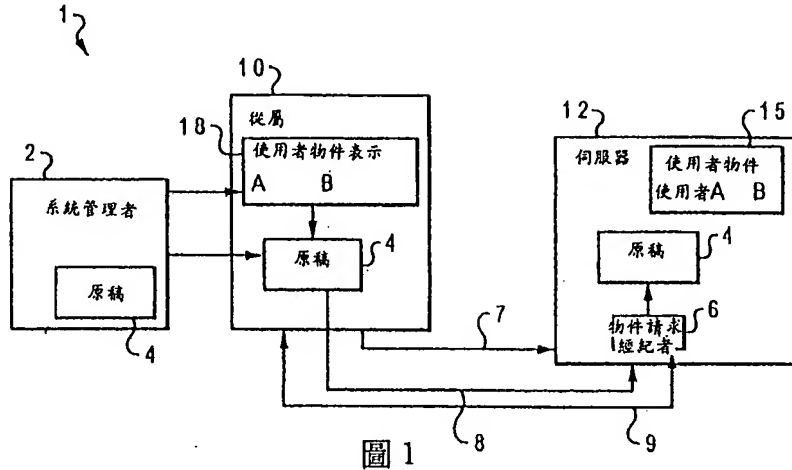


圖 1

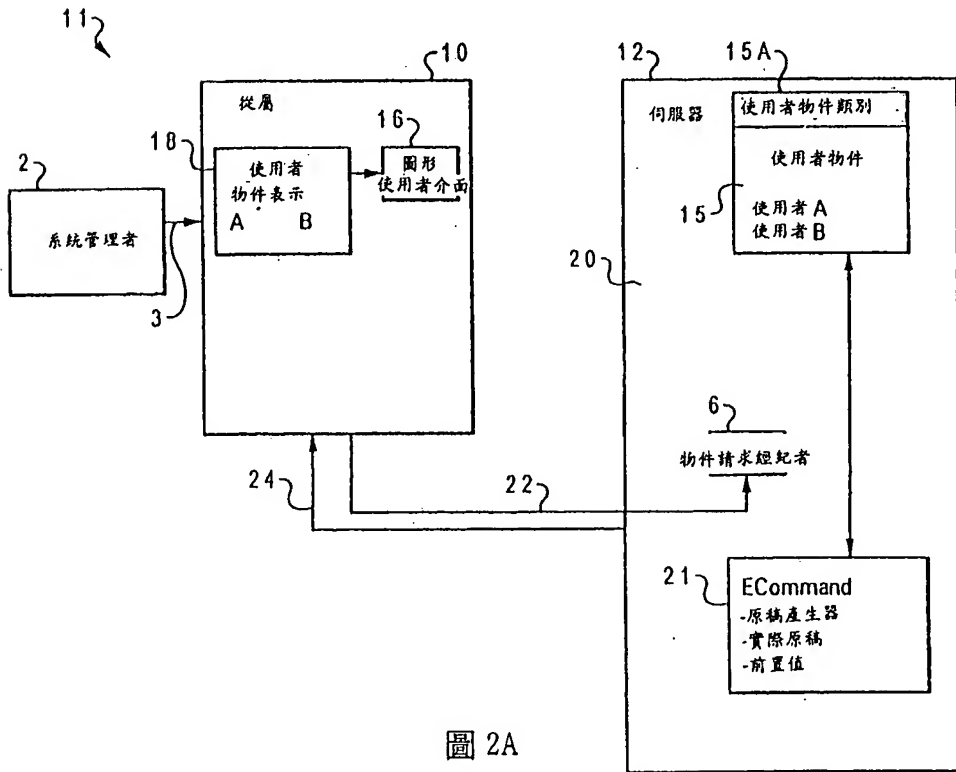


圖 2A

(5)

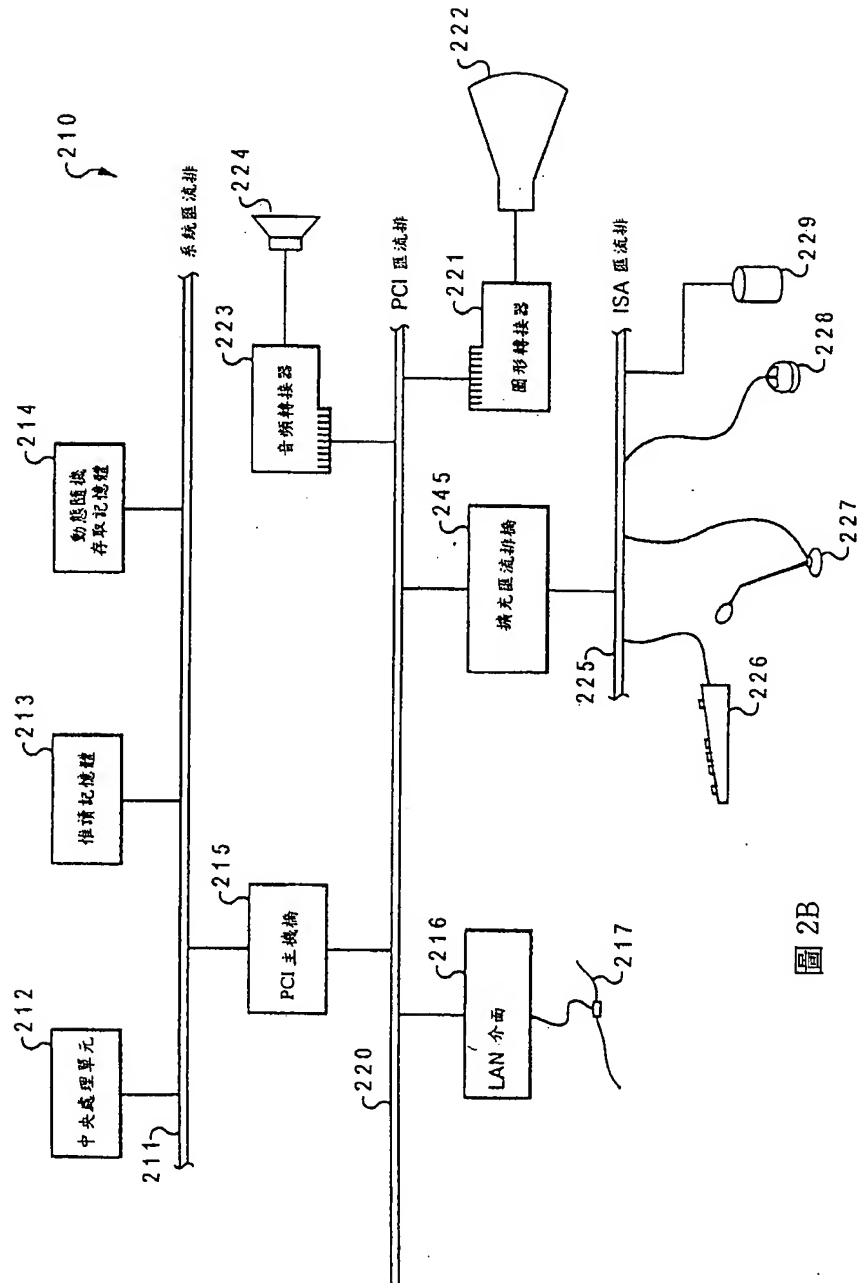


圖 2B

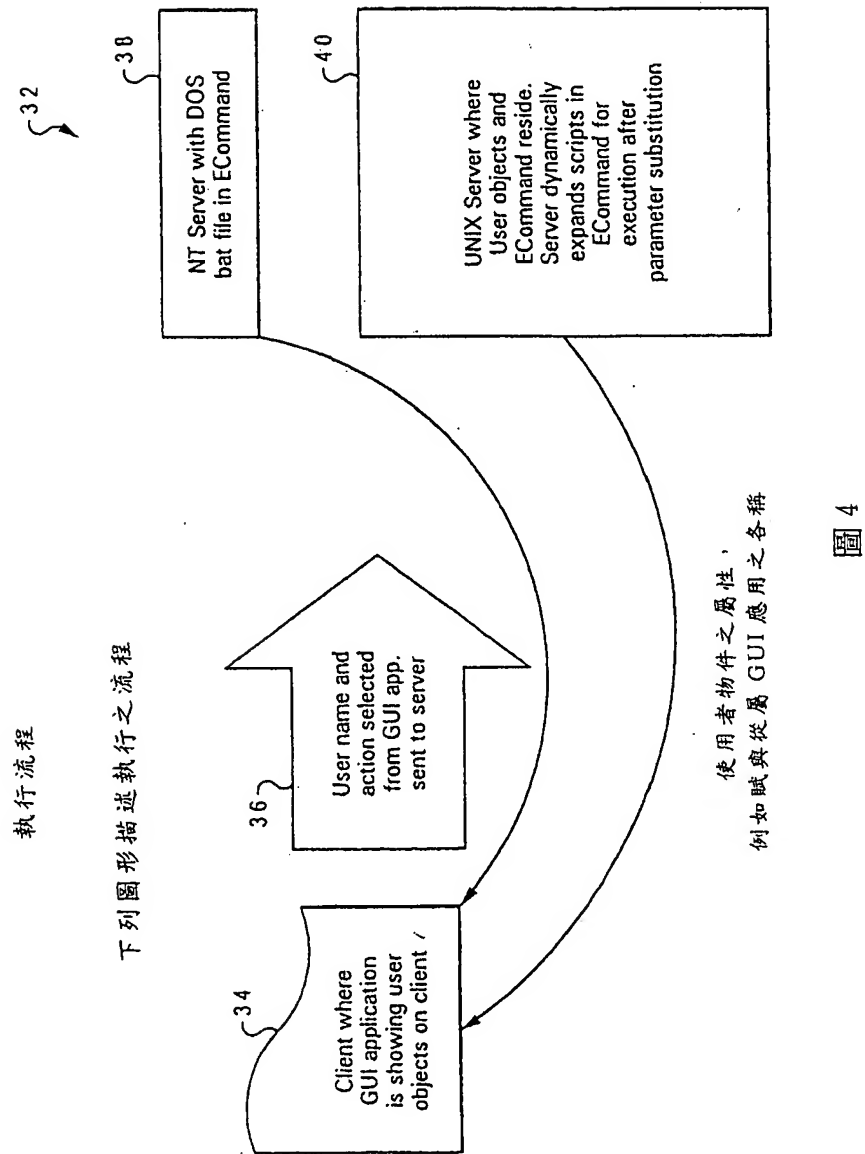
(6)

26 ↘  
28 ↘ 27 ↘ 29 ↘ 27 ↘ 29 ↘  
ECommand stanzaName (Object Type prefix 1, Object Type prefix 2, .....)  
{ Method = {script code goes here.....} } (for one Operating System) 30  
{ Method = {script code goes here.....} } (for another Operating System) 30  
{ Method = {script code goes here.....} } (for a further Operating System) 30  
其中  
- stanzaName 是節之名稱  
- Object Type 是傳送至 ECommand 物件之物件型式  
- prefix # 是變數之前置

圖 3

80 ↘  
User Script.cmd  
package com.ibm.websm.test.cmd; 84  
import java.util.\*;  
ECommand delete (MObject User, Hashtable Action)  
{  
Method = {  
28 A ↘ 87 ↘ 83 ↘ 85 ↘  
## Delete a user  
if [ [ \${Action\_Name} = "Force\_Delete" ] ]; then  
rmuser \${User\_Name} 85  
elif [ [ \${User\_Name} = "root" ] ]; then  
echo "Cannot delete root" 86  
81 ↘ 81 ↘  
elif [ [ \${User\_Deletable} = "false" ] ]; then  
echo "You are not permitted to delete this user"  
else  
rmuser \${User\_Name}  
fi  
}  
}

圖 5A



```

// A very simple UserObject class representing
// user on a UNIX System
public class UserObject extends MObject
{
    // Constructor for UserObject
    public UserObject (String name)
    {
        _name = name;
    }
    // Method to delete the user
    public void delete ()
    {
        Hashtable Action = new Hashtable ();
        Action.put ("Name", "");
        String cmd = userScript.delete.expand(this, Action);
        // run the script using Java exec call
        Runtime.getRuntime().exec(cmd);
    }
    // Method to force delete the user
    public void forceDelete ()
    {
        Hashtable Action = new Hashtable ();
        Action.put ("Name", "Force_Delete");
        String cmd = userScript.delete.expand(this, Action);
        // run the script using Java exec call
        Runtime.getRuntime().exec(cmd);
    }
    // Method to getPropertyValues of a User object
    public Hashtable getPropertyValues ()
    {
        Hashtable pValues = new Hashtable ();
        pValues.put ("Name", _name);
        pValues.put ("Deletable", isDeletable());
        return pValues;
    }
    // Method to determine if user is deletable
    public String isDeletable()
    {
        if(_name.equals ("root") )
            return "false";
        return "true";
    }
    private String _name;

```

15 A

90

28 A

28 B

28 C

28 D

圖 5B



## User Script. JAVA

```
package com.ibm.websm.test.cmd;
import java.util.*;
```

94

```
import com.ibm.websm.etc*;
```

```
public class userScripts extends ECommandGroup {

    public static ECommand delete = new ECommand ();

    static String [] delete_prefixnames = {
        "User",      ← 8 4 A
        "Action",    ← 8 4 B
    };

    static String [] delete_varlist_User = {
        "Name",      ← 8 5
        "Deletable", ← 8 6
    };

    static String [] delete_varlist_Action = {
        "Name",      ← 8 7
    };

    static String delete_script =
        "\n" +
        "\n" +
        "## Delete a user\n" +
        "if [ [ ${Action_Name} = \"Force_Delete\" ] ]; then\n" +
        "rmuser ${User_Name}\n" +
        "elif [ [ ${User_Name} = \"root\" ] ]; then\n" +
        "echo \"Cannot delete root\"\n" +
        "elif [ [ ${User_Deletable} = \"false\" ] ]; then\n" +
        "echo \"You are not permitted to delete this user\"\n" +
        "else\n" +
        "rmuser ${User_Name}\n" +
        "fi\n" +
        "\n" +
        " ";

    static {
        String [ ] [ ] delete_attributeLists = new String [2] [ ];
        delete_attributeLists [0] = delete_varlist_User;
        delete_attributeLists [1] = delete_varlist_Action;

        delete = new ECommand(
            delete_script,
            delete_prefixnames,
            delete_attributeLists);
    } /* end static */
}
```

圖 5C

```

##— Begin Variables —##
87 VimName='default'
    VimCLASS='default' 89
85 User_Name='dummyUser' 96
86 User_Deletable=' '
    Action_Name='Force_Delete'

##— End Variables —##

## Delete a user 87
if [ [ ${Action_Name} = "Force_Delete" ] ]; then
    rmuser ${User_Name} 85
elif [ [ ${User_Name} = "root" ] ]; then
    echo "Cannot delete root" 86
elif [ [ ${User_Deletable} = "False" ] ]; then
    echo "You are not permitted to delete this user"
else
    rmuser ${User_Name}
fi

```

圖 5D

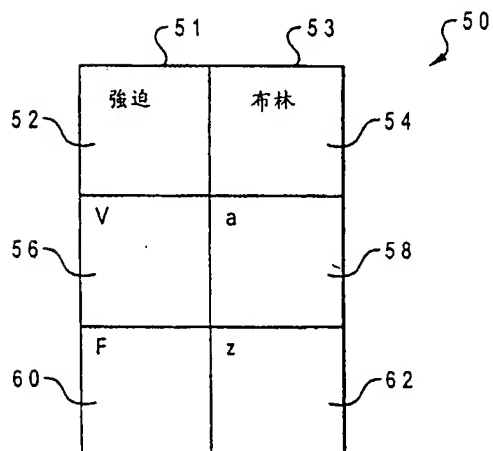


圖 6

ECommand 擴展方法之流程圖

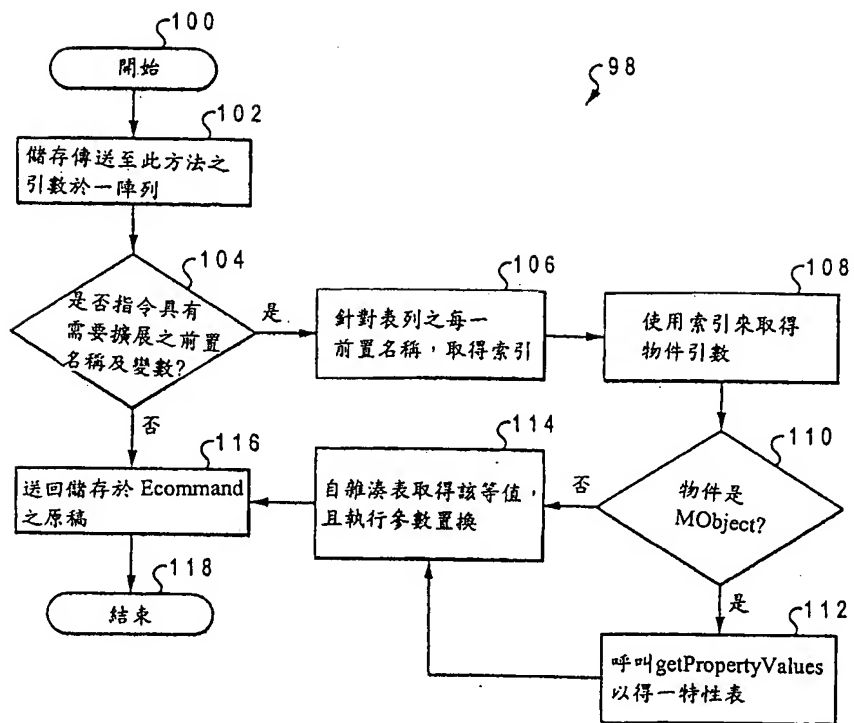


圖 7

